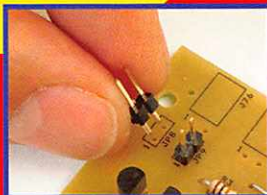
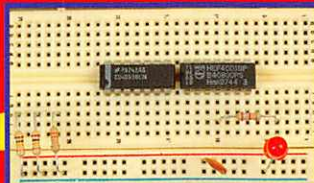
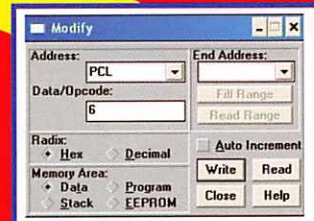
...e costruisci il tuo **LABORATORIO DIGITALE**

6,90 €

29

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41029

Peruzzo & C.

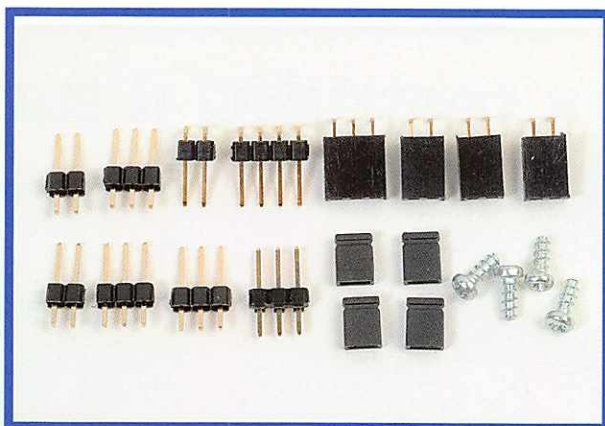
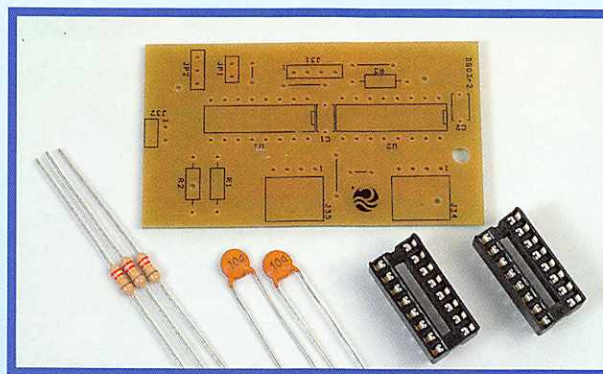
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE" si compone di 70 fascicoli settimanali da suddividere in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

- 3 Connettori femmina da c.s. a due vie a 90°
- 1 Connettore femmina da c.s. a tre vie a 90°
- 1 Connettore maschio da c.s. a due vie a 90°
- 1 Connettore maschio da c.s. a quattro vie a 90°
- 4 Connettori maschio da c.s. a tre vie diritti
- 2 Connettori maschio da c.s. a due vie diritti
- 4 Ponticelli isolati
- 4 Viti

IN REGALO nel prossimo fascicolo

- 1 Scheda DG03r2
- 2 Zoccoli DIL 16 pin
- 2 Condensatori ceramici 100 nF
- 3 Resistenze 220 K 5% 1/4 W

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

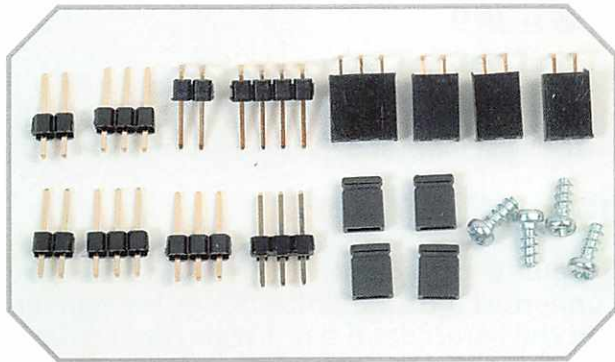
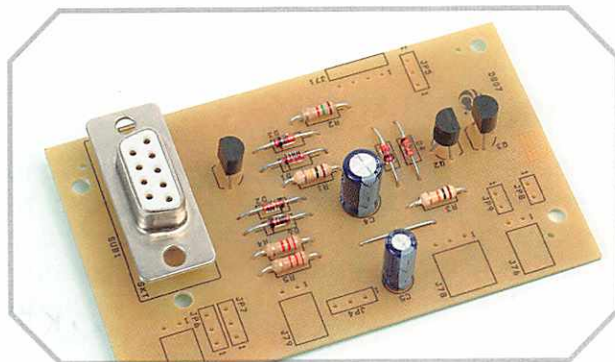
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:
elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

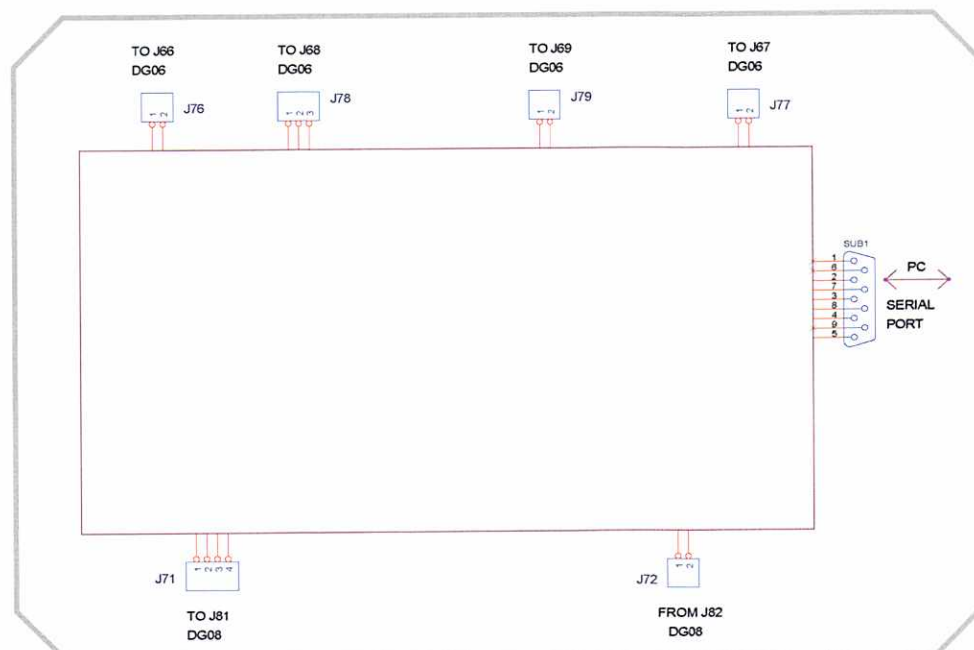
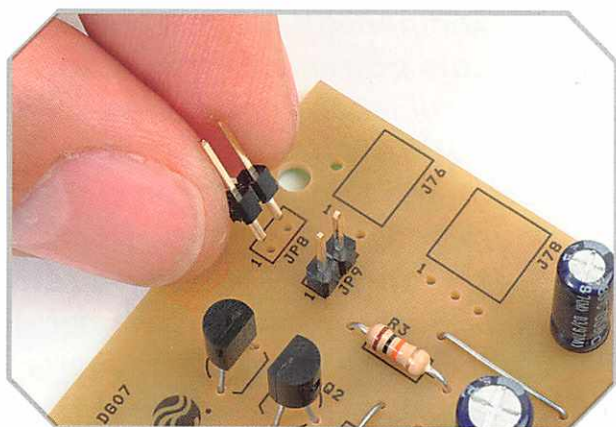
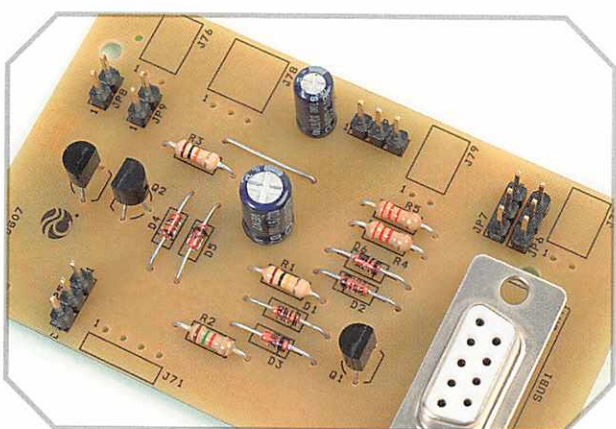
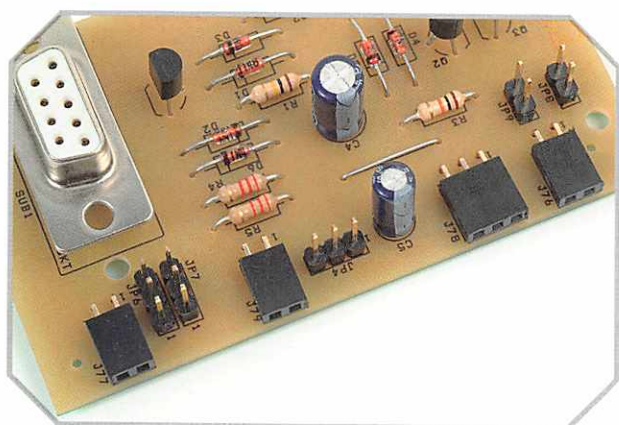
Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Circuito di scrittura (III)

Componenti forniti con questo fascicolo.

Situazione attuale della scheda DG07.

Con i componenti allegati a questo fascicolo potremo terminare e installare sul laboratorio la scheda di scrittura DG07. Vengono forniti i connettori per i ponticelli con cui si configura la scheda e i quattro ponticelli i connettori attraverso cui la scheda si collegherà a quella del PIC – DG06 – e la scheda di memoria DG08. Vengono fornite anche le quattro viti che saranno utilizzate per fissare la scheda al laboratorio.

Montaggio

Osservando il circuito stampato vedremo che manca solamente l'installazione di alcuni connettori. Li installeremo uno per uno, inserendo i loro terminali nei fori corrispondenti e saldandoli successivamente con molta cura, applicando il saldatore il tempo sufficiente per ottenere la temperatura adeguata sia per lo stagno che per il componente.

Schema elettrico ingressi e uscite.

Connettori per i ponticelli di configurazione.

Montaggio dei connettori
a tre vie per i ponticelli di coinfigurazione.

Connettori femmina
a 90° per i collegamenti alla scheda del PIC.

JP8 e JP9

I primi componenti con cui riprenderemo il montaggio sono i due connettori a due vie diritti, che si montano sulle posizioni JP8 e JP9.

JP5, JP6, JP7 e JP8

Su questi riferimenti si montano i quattro connettori a tre vie diritti. Sia questi connettori che i precedenti si utilizzano per configurare le diverse funzioni della scheda grazie a dei ponticelli, in questo caso vengono forniti quattro ponticelli, ricordando che in precedenza ve ne sono stati forniti altri per la realizzazione di ogni esercizio sino a qui sperimentato.

Collegamenti a DG06

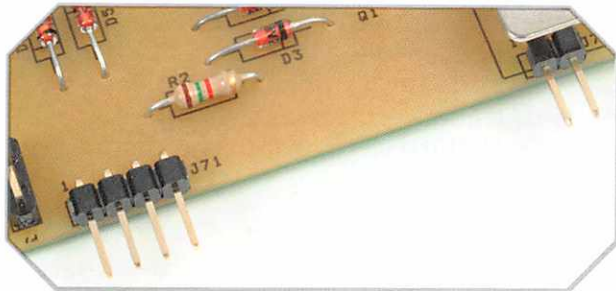
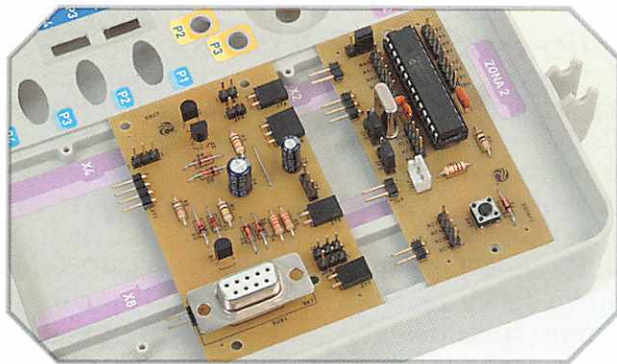
I collegamenti alla scheda DG06 si eseguono tramite i connettori J76, J77, J78 e J79, l'ultimo dei quali permette di alimentare la scheda DG06 in funzionamento normale tramite questa scheda, cioè tramite la DG07. Questi connettori sono di tipo femmina e piegati a 90°, J76, J77 e J79 sono a due vie, mentre J78 è a tre vie.

È molto importante che questi connettori rimangano ben allineati per facilitare il loro collegamento alla scheda DG06, a questo scopo devono essere perfettamente appoggiati sulla scheda e perpendicolari alla stessa nel momento della saldatura.

Collegamenti a DG08

I collegamenti a DG08 si eseguono con due connettori, il J71 a quattro vie, per i collegamenti alla scheda di memoria, e il connettore J72, attraverso il quale la scheda riceve l'alimentazione tramite la DG08. Il terminale 1 corrisponde al negativo e il terminale 2 al positivo da 5 V; questi terminali sono di tipo maschio e piegati a 90°, e devono essere ben allineati prima di effettuare le saldature.

Revisioni

Bisogna rivedere molto bene tutto il lavoro di montaggio realizzato sulla scheda, prima di installarla sul laboratorio, in quanto un errore

Connettori maschio a 90° per i collegamenti alla DG08.

L'allineamento dei connettori facilita la connessione.

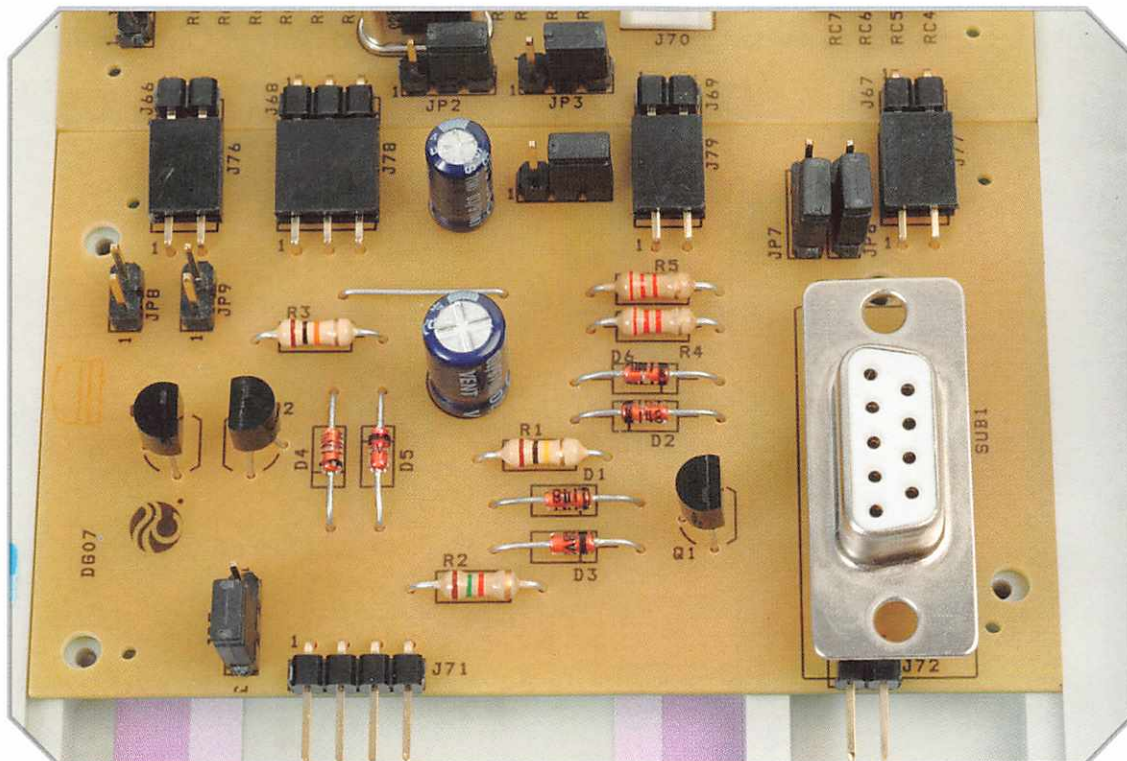
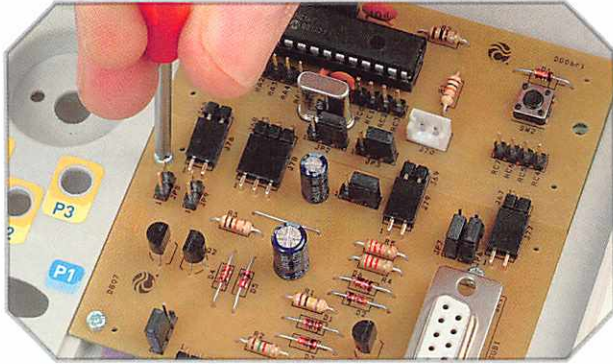
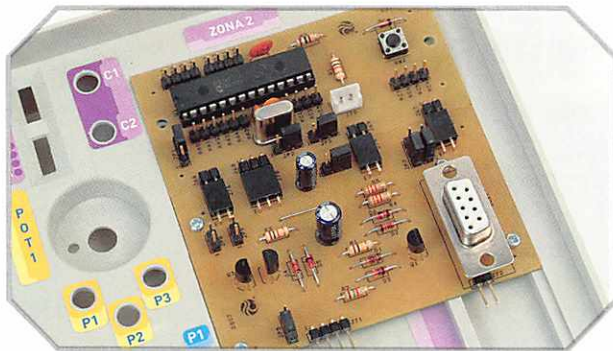
nella scheda può causare la distruzione del PIC.

Le verifiche che vi sono state indicate nei fascicoli precedenti, dovrebbero già essere state eseguite, altrimenti dovete farlo ora, prima di installare la scheda. È necessario controllare la posizione e l'orientamento di ogni componente, e che il componente corrisponda esattamente al riferimento. È importante verificare di aver effettuato ogni saldatura e che non ci siano cortocircuiti di stagno nelle zone dove i terminali sono molto vicini. Questa scheda ha un ponticello di filo nudo, indicato con un tratto continuo, vicino alla resistenza R3, che non deve essere dimenticato.

Montaggio di DG07

Come vi è stato indicato, questo circuito viene montato fra il circuito che contiene il PIC e quello che si utilizzerà per supportare il dispositivo lettura/scrittura della scheda di memoria.

Questo circuito si monta sul laboratorio fissandolo con le quattro viti che vi sono state fornite, alle quattro torrette siglate come X2, X3, X6 e X7 nella zona 2 del laboratorio.

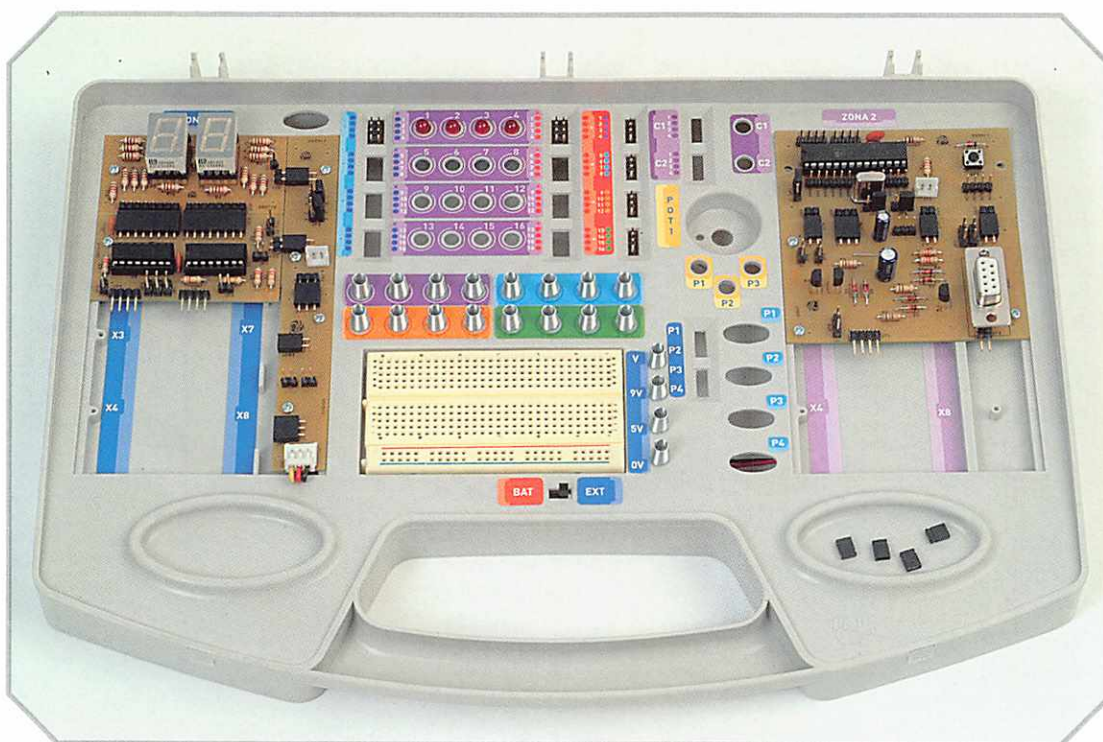
La scheda deve essere centrata, come si può

Scheda DG07 completa.

La scheda DG07 si fissa con quattro viti.

Schede DG06 e DG07 montate.

vedere nelle fotografie, con il connettore a 9 terminali per il collegamento al PC orientato verso il bordo destro del laboratorio. Le viti si devono chiudere ma senza stringere eccessivamente, infatti va ricordato che stiamo avvitando su della plastica.

Montaggio di DG06

La scheda DG06 si fissa temporaneamente, utilizzando i connettori che la uniscono alla DG07, più avanti verranno fornite le viti per il suo fissaggio.

Alimentazione di DG06

La scheda DG06 può alimentarsi tramite la DG07, in funzionamento normale. Sino a quando non vi verrà fornita la DG08, si utilizzerà il connettore J72 della scheda DG07 come alimentazione, ricordando ancora una volta che il terminale 1 è il negativo e il 2 il positivo da 5 V. Quando si scrive il PIC l'alimentazione si prende dal PC, non è necessario quindi utilizzare l'alimentazione del laboratorio, anche se bisogna tener presente la configurazione dei ponticelli sulle schede DG06 e DG07 per i diversi modi di utilizzo di queste schede.

Laboratorio con la scheda DG07 inserita.

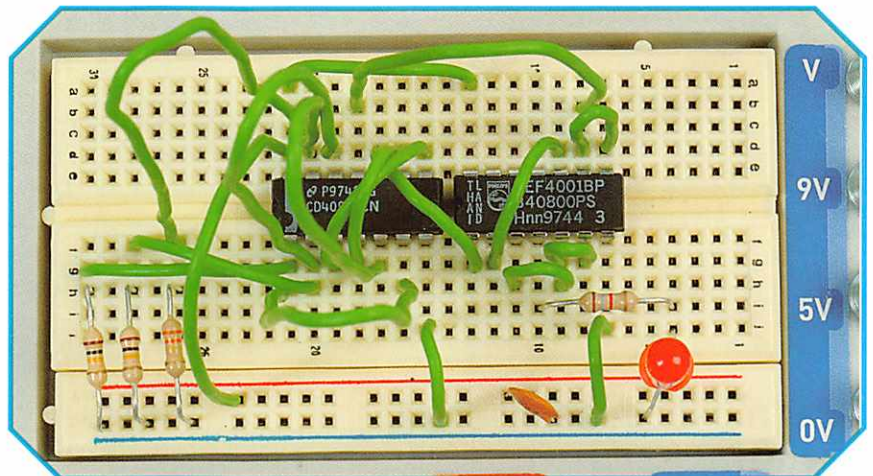
Funzione logica maggiore

La funzione maggiore indica che la maggior parte delle variabili, in questo caso tre, sono a livello alto.

Il circuito

Osservando lo schema vedremo che si utilizza una combinazione di porte NAND e NOR in modo da ottenere in uscita la funzione che rappresentiamo sulla tabella della verità. Spiegare le otto combinazioni possibili e seguire il segnale da ogni ingresso verso l'uscita sarebbe molto noioso, tuttavia, come esercizio, converrebbe scegliere qualche combinazione di variabili di ingresso e seguire lo schema fino a ottenere il valore in uscita.

È sicuramente più rapido montare l'esperimento e seguire tutte le combinazioni possibili con i collegamenti di ingresso ABC, tenendo presente che quando il collegamento non è realizzato l'ingresso è 0 e quando lo è l'ingresso vale 1.

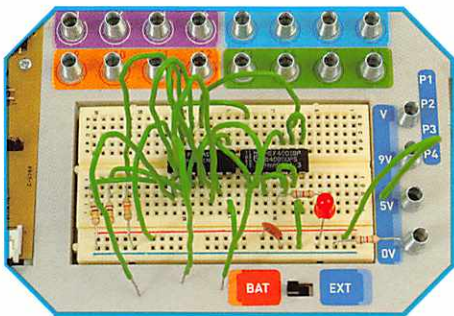
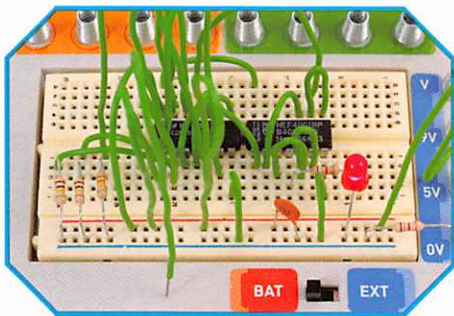
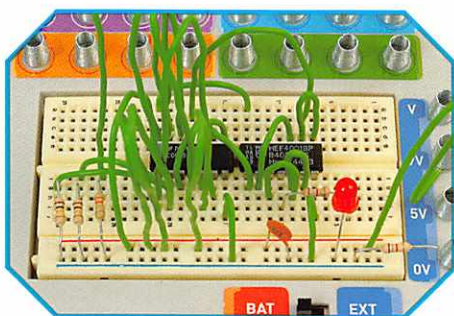
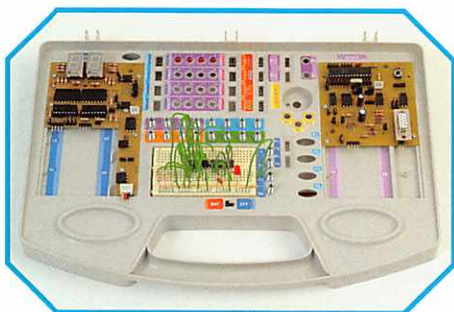
Componenti sulla scheda Bread Board.

Cablaggio dei componenti dell'esperimento.

Basso consumo

Gli integrati CMOS hanno un consumo molto ridotto, tanto che la corrente di alimentazione è molto piccola, il circuito funziona anche inserendo una resistenza in serie all'alimentazione, cosa che si può fare sul positivo o sul negativo; in questa prova inseriremo la resistenza R5, verificando che il circuito continui a funzionare. Potremo verificare inoltre che il circuito funziona collegando l'alimentazione con un filo, invece di inserire la resistenza.

Per quanto concerne il condensatore C1, esso filtra la tensione di alimentazione dai possibili disturbi della linea di alimentazione, specialmente quelli a frequenze molto elevate. Alimentando a pile e in circuiti più semplici, normal-

Funzione logica maggiore.

Collegamenti della alimentazione a 9 V con una resistenza inserita sul negativo.

Montaggio completo.

Con A e B chiusi l'uscita è 1.

Con A, B e C chiusi, l'uscita è ancora 1.

Laboratorio con l'esperimento realizzato.

mente non è necessario questo tipo di filtro.

Funzionamento

Il funzionamento del circuito una volta montato, deve seguire la tabella della verità, e quando la maggior parte delle variabili è a 1, l'uscita è 1, questo è indicato attraverso l'illuminazione del LED A.

Funzione maggiore

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Montaggio

Il montaggio di questo esperimento si realizza come d'abitudine, iniziamo dall'inserimento dei due integrati utilizzati sulla scheda Bread Board e successivamente il resto dei componenti, facendo attenzione all'orientamento degli integrati e alla polarità del LED A. In seguito si realizza il cablaggio interno alla scheda, seguendo lo schema per non dimenticare nessuna connessione, e dopo aver verificato la corretta esecuzione del lavoro potremo collegare l'alimentazione.

Alimentazione

Dopo aver verificato che il montaggio è stato realizzato in modo corretto si collega l'alimentazione, il negativo di questa è sulla molla 0 V e il positivo a 9 V. Sostituiremo uno dei fili di collegamento dell'alimentazione con una resistenza da 1K8 e vedremo che il circuito funziona. Il commutatore di alimentazione deve essere sulla posizione BAT.

LISTA DEI COMPONENTI

U1	Circuito integrato 4093
U2	Circuito integrato 4001
R1, R2	Resistenza 100 K (marrone, nero, giallo)
R3	Resistenza 330 K (arancio, arancio, giallo)
R4, R5	Resistenza 1K8 (marrone, grigio, rosso)
C1	Condensatore 22 nF
LED A	Diodo LED rosso 5 mm

Terzo esperimento con il PIC16F870

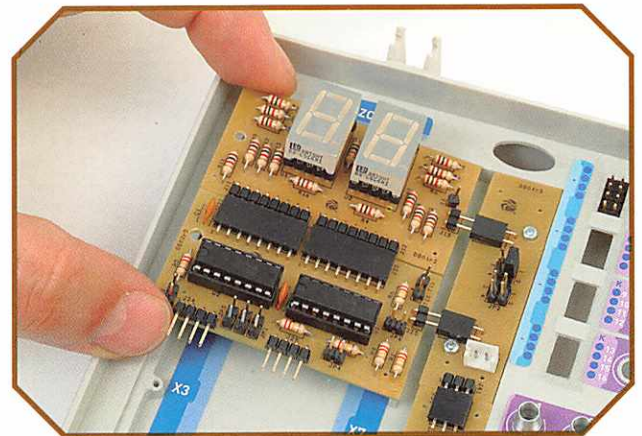
Continueremo con il primo progetto, il cui programma è fornito già scritto nel PIC, e di cui conosciamo già la struttura.
L'illuminazione dei segmenti del display è controllata direttamente dal PIC, senza la necessità di utilizzare il driver.

Utilizzeremo l'altra parte del programma, impiegando in questo caso come uscite i terminali da RB0 a RB7 della porta B, e il terminale RA0 della porta A per il controllo.

Collegamento diretto al display

Un altro modo di risolvere la visualizzazione dei numeri in un display a 7 segmenti, consiste nel collegare quest'ultimo direttamente alle uscite del PIC, senza ricorrere all'utilizzo dei driver.

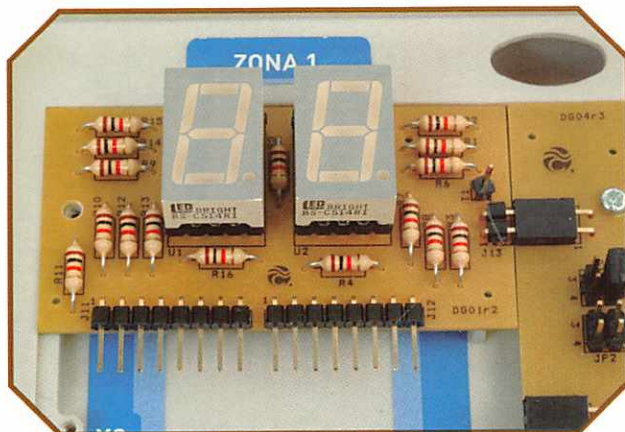
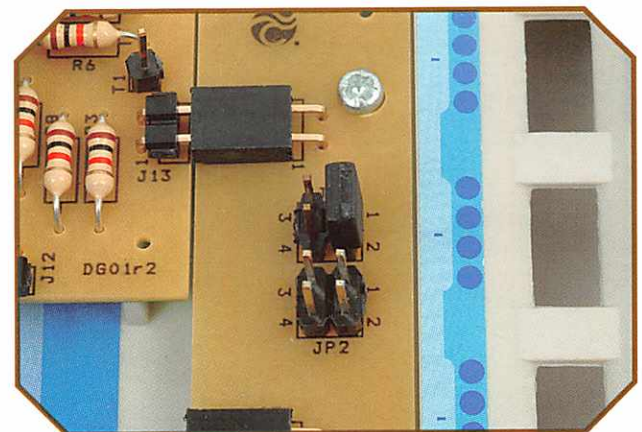
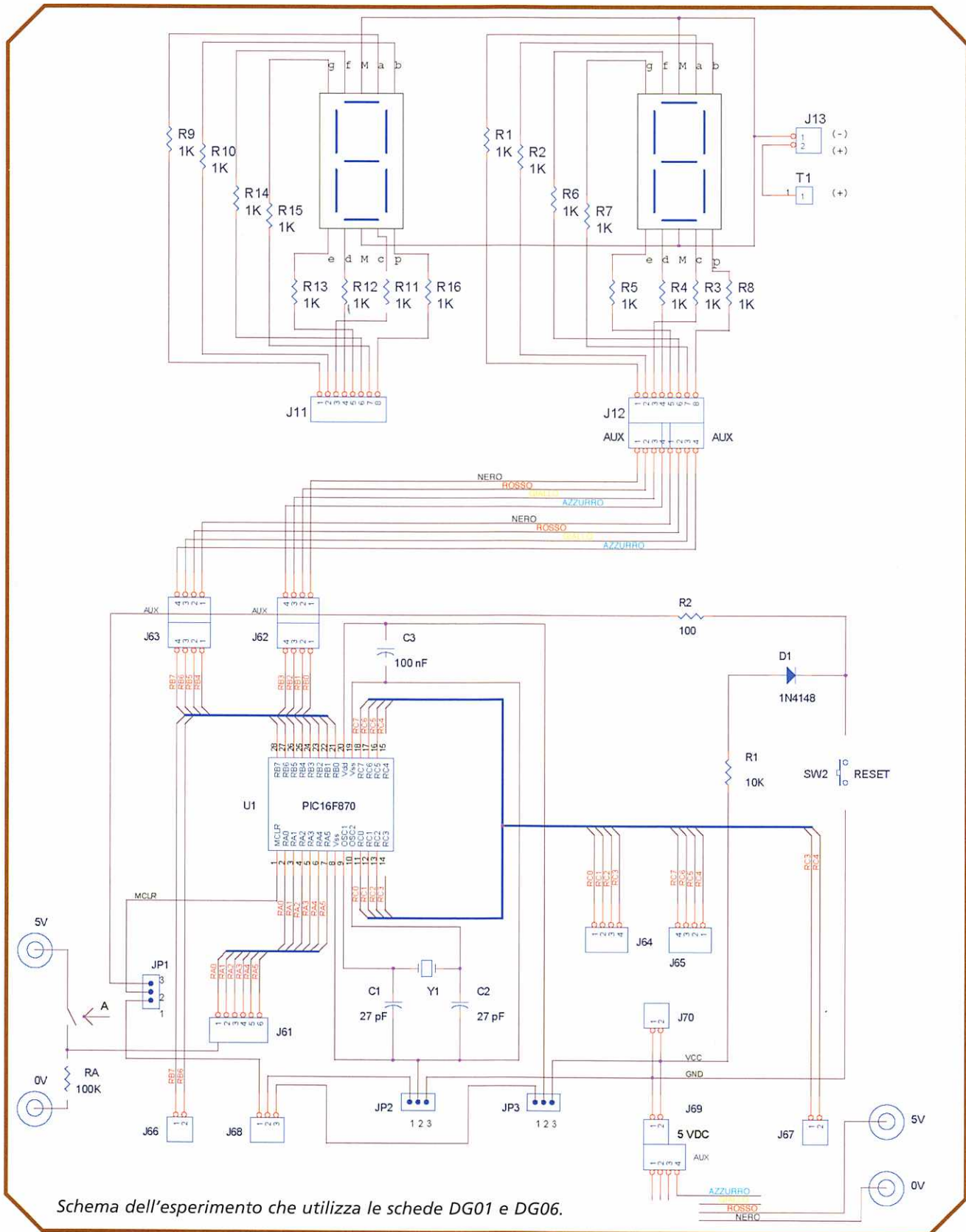
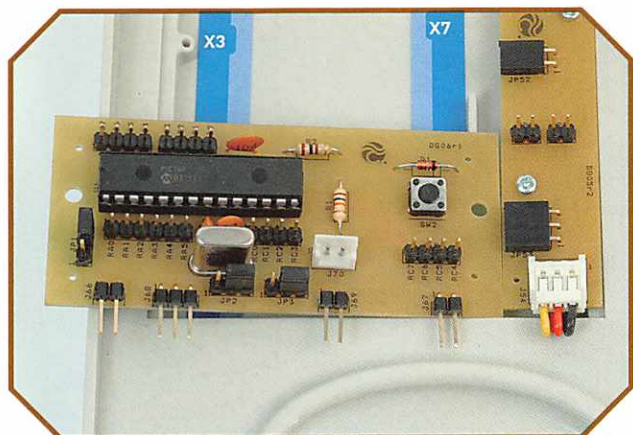
Conosciamo già la potenza del PIC, quindi ci domandiamo se il microcontroller può realizzare la funzione del driver e fornire le uscite direttamente in codice a 7 segmenti. Con il PIC 16F870 possiamo sostituire il driver e rappresentare un numero in codice a 7 segmenti sul display. L'unica cosa di cui abbiamo bisogno è una porta di uscita libera e creare il codice in assembler che realizzi la conversione.

Per quanto riguarda il codice non è molto difficile creare una subroutine che risolva questa conversione, e in merito alla porta di uscita abbiamo disponibile la Porta B, quindi la configureremo nel programma di uscita e la utilizzeremo per collegare direttamente il display.

Bisogna estrarre la scheda DG02.

È importante notare che il programma contenuto nel microcontroller contempla questa opzione e fornisce sulle uscite della Porta B il numero da rappresentare in codice a 7 segmenti.

Il montaggio

Il montaggio di questo esercizio si può dedurre praticamente dalle illustrazioni, però bisogna eseguire alcune preparazioni preventive.

Lasceremo libero l'accesso ai connettori del-

Scheda DG01 con i connettori accessibili.

La posizione dei ponticelli della scheda DG04 in questo esercizio è indifferente.

Schema dell'esperimento che utilizza le schede DG01 e DG06.

La Scheda DG06
si deve posizionare vicino alla DG01.

Collegamento del primo
cavetto fra la porta B e la scheda DG01.

la scheda DG01, togliendo le viti che fissano le schede DG01 e DG02 e staccando entrambe dalla scheda di alimentazione DG04. Dopo aver scollegato le schede DG01 e DG02 dalla DG04, le separeremo, riporremo la scheda DG02 e inseriremo il connettore J13 di DG01 sul connettore J42 di DG04.

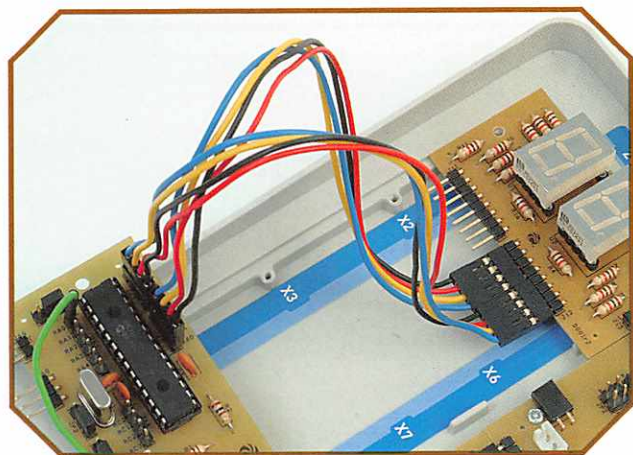
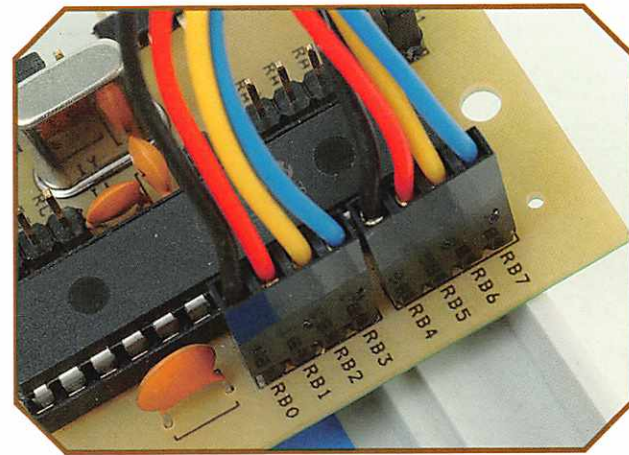
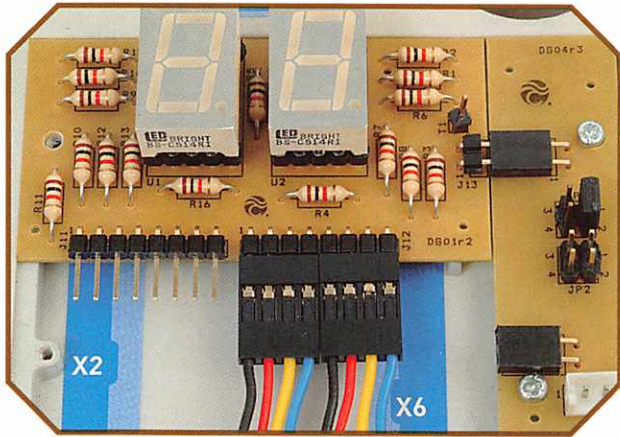
Continueremo togliendo dal suo alloggiamento la scheda DG06, che ricordiamo non è ancora stata avvitata.

I terminali della porta B si trovano sulla parte superiore della scheda DG06 che contiene il microcontroller e sono raggruppati di quattro in quattro. In questo modo si facilita il collegamento dei due cavetti a quattro fili.

Con uno dei cavetti si unisce il connettore J62 ai terminali da 1 a 4 in J12 della scheda DG01, tenendo presente l'ordine seguito dai colori. L'altro cavetto unisce il connettore J63 di DG06 ai terminali dal 5 all'8 del connettore J12.

È molto importante non incrociare i cavetti e collegarli in modo adeguato, altrimenti la rappresentazione sul display non sarà quella desiderata e appariranno caratteri strani senza alcun significato. Potete provare a incrociare i cavetti e provare così la visualizzazione risultante.

È necessario inoltre far attenzione a non appoggiare la scheda DG06 sopra zone che potrebbero provocare un cortocircuito, evitando quindi di posarla su oggetti metallici.

Il connettore J12 si utilizza per i collegamenti dei cavetti. Lo stesso risultato si otterrebbe utilizzando il connettore J11 e l'altro display.

Collegamento del secondo
cavetto fra la porta B e la scheda DG01.

Collegamento alla porta B,
bisogna fare attenzione all'ordine dei colori.

Collegamento a uno dei display.

Dettaglio del circuito di controllo, collegato alla porta A, terminale RA0.

Ingresso di controllo

In questo esperimento si utilizza lo stesso ingresso di controllo dei precedenti, il terminale 0 della porta RA e una resistenza da 100 K. Quando l'ingresso RA0 è a livello "1", si visualizzano i numeri in modo sequenziale e passando a livello "0" viene visualizzato il numero casuale ottenuto.

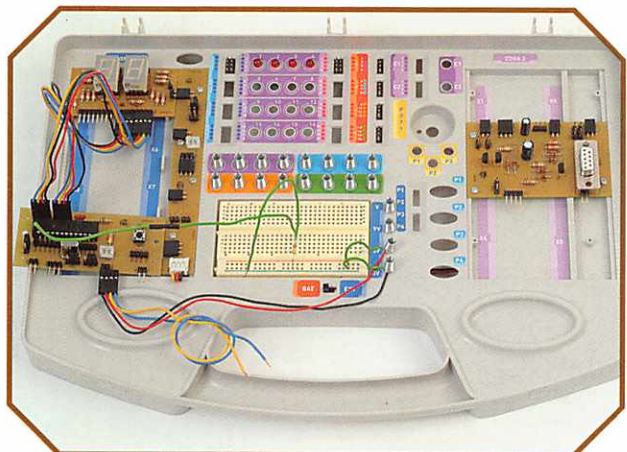
Alimentazione

L'alimentazione della scheda DG06 si prende dalle molle 0 V e 5 V. Si utilizza un cavetto a quattro fili, dei quali ne impiegheremo solamente due, il rosso per il positivo collegato a 5 V e il nero per il negativo collegato a 0 V come GND. La scheda DG01 si alimenta tramite la DG04 e prende da questa solamente il negativo, quindi è indifferente che siano collocati o meno i ponticelli della scheda DG04. Infine il commutatore di alimentazione deve essere sulla posizione BAT.

Conclusioni

Mediante questo esercizio si verifica la versatilità e la flessibilità del laboratorio e del microcontroller. Esistono molti modi per risolvere un progetto, anche se spesso le limitazioni economiche ci indirizzeranno verso un percorso determinato.

Giocate con questo montaggio, ma facendo sempre molta attenzione, in quanto anche se si lavora con tensioni o con correnti molto basse che non possono normalmente causare danni personali, è possibile danneggiare i circuiti.

L'alimentazione di DG06 è di 5 V.

Laboratorio con l'esperimento realizzato.

Programma per i primi esperimenti con il PIC (II)

Continuiamo con il programma per il nostro progetto, seguendo però la logica sequenza di sviluppo di un progetto.

Dopo aver realizzato il codice, dobbiamo verificare che questo venga compilato senza errori e che risponda all'enunciato mediante la simulazione.

Preparazione del codice

Lo strumento da utilizzare per assemblare e compilare il nostro codice è MPLAB, però prima di compilare dobbiamo "preparare" il nostro programma. Con il termine preparare intendiamo fare riferimento al lavoro di messa a punto e depurazione per rendere il nostro programma veramente pronto. A questo scopo dobbiamo ordinare il codice, rivederlo e aggiungere i commenti per rendere più semplice la sua comprensione. È bene quindi abituarci a inserire delle intestazioni in cui vengano spiegate le funzioni realizzate da ogni blocco differenziato del programma. Possiamo intestare il programma con l'enunciato iniziale dell'applicazione, ogni subroutine con la funzione che realizza, e perché no, il modo in cui è stata risolta. Possiamo inoltre separare – come abbiamo fatto quando abbiamo spiegato il codice – le parti di programma più caratteristiche o che realizzano funzioni specifiche. Nelle illustrazioni viene evidenziato come un progettista lascia il suo codice totalmente "sistemato".

Fatto questo, ordinato e rivisto tutto il codice, possiamo procedere alla sua compilazione.

MPLAB IDE - C:\PROGRA~1\MPLAB\PROGETTI\ED28.PJT - [c:\progra~1\mplab\progetti\ed28.asm]

File Project Edit Debug PICSTART Plus Options Tools Window Help

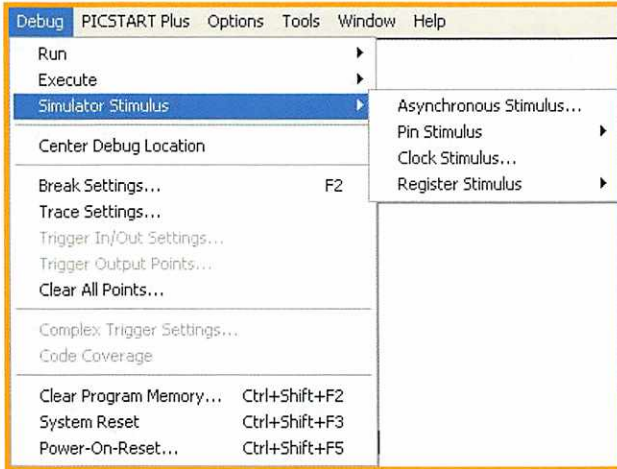
```
;Generazione di numeri casuali.
;------------------------------
;Generazione di un numero casuale fra 1 e 9. Quando RA0 è a "1", sul display a 7
;segmenti collegato alla porta B si visualizzeranno sequenzialmente i numeri da 1 a 9
;e su i LED collegati alla porta C lo stesso numero però in binario. Questo avviene
;a intervalli di 0,05". Quando RA0 passa a livello "0", visualizziamo il numero casuale
;ottenuto per un tempo di 3". Dopo, il display e i LED si spengono e la sequenza si ripete.

                List    p=16F870            ;Processore
                include "P16F870.INC"       ;Definizioni dei registri interni

Numero          equ     0x20                ;Numero casuale
Delay_Cont      equ     0x21                ;Contatore di intervalli
Temporale       equ     0x22                ;Variabile temporale

                org     0x00                ;Vector di Reset
                goto    Inizio
                org     0x05                ;Salva il vector di interrupt
;-------------------------------------------------------------------------------
;Tabella: Routine che converte il codice binario presente sui 4 bit meno significativi
;del registro W nel loro equivalente a 7 segmenti. Il codice a 7 segmenti ritorna su W

Tabella:        addwf   PCL,F               ;Spostamento sulla tabella
                retlw   b'00111111'         ;Numero 0
                retlw   b'00000110'         ;Numero 1
                retlw   b'01011011'         ;Numero 2
                retlw   b'01001111'         ;Numero 3
                retlw   b'01100110'         ;Numero 4
                retlw   b'01101101'         ;Numero 5
                retlw   b'01111101'         ;Numero 6
                retlw   b'00000111'         ;Numero 7
                retlw   b'01111111'         ;Numero 8
                retlw   b'01100111'         ;Numero 9

;-------------------------------------------------------------------------------
;Delay_20_ms: Routine di temporizzazione per eliminare "l'effetto rimbalzo" dei dispositivi
;elettromeccanici. Realizza un ritardo di 20 ms. Il PIC lavora ad una frequenza di 4 MHz,
;il TMR0 conta ogni µs. Se vogliamo temporizzare 20000 µs (20 ms) con un prescaler di 256,
;il TMR0 deve contare 78 eventi (78 * 256). Il valore 78 equivale a 0x4e hex. e dato che il
;TMR0 è ascendente dovremo caricare il suo complemento a 1 (0xb1 hex.).

Delay_20_ms:    bcf     INTCON,T0IF         ;Azzera il flag di overflow
```

Ln 35 Col 1 | 139 | # | WR | No Wrap | INS | PIC16F870 | pc:0x00 | w:0x00 | --z dc c | Bk On | Sim | 4 MHz | User

Il codice sta assumendo la sua forma man mano che si avanza nella preparazione.

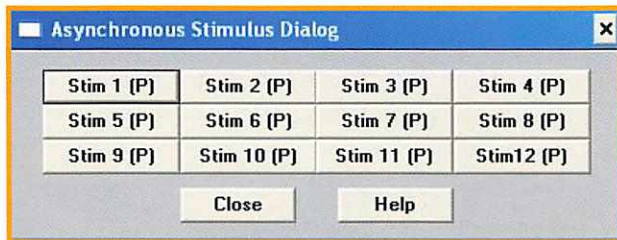
Build Results

```
Building ED28.HEX...

Compiling ED28.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\ED28.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ED28.ASM 63 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ED28.ASM 64 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ED28.ASM 66 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ED28.ASM 68 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ED28.ASM 71 : Register in operand not in bank 0.  Ensure

Build completed successfully.
```

Risultato della compilazione e assemblaggio.

Menù per accedere alla simulazione degli ingressi.

Asynchronous Stimulus Dialog

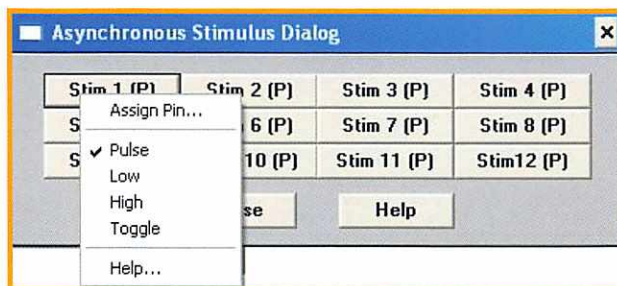
Stim 1 (P)	Stim 2 (P)	Stim 3 (P)	Stim 4 (P)
Stim 5 (P)	Stim 6 (P)	Stim 7 (P)	Stim 8 (P)
Stim 9 (P)	Stim 10 (P)	Stim 11 (P)	Stim12 (P)

Close Help

Finestra per stimolare stimoli asincroni.

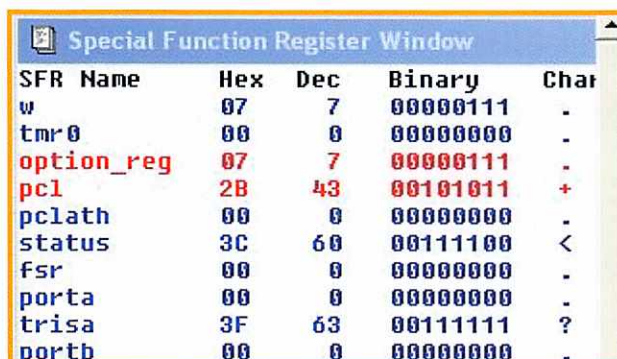
Asynchronous Stimulus Dialog

Stim 1 (P) | Stim 2 (P) | Stim 3 (P) | Stim 4 (P)
S | 6 (P) | Stim 7 (P) | Stim 8 (P)
S | 10 (P) | Stim 11 (P) | Stim12 (P)

Assign Pin...
✔ Pulse
Low
High
Toggle
Help...

se Help

Per assegnare un ingresso clicchiamo con il pulsante destro del mouse sul comando.

Special Function Register Window

SFR Name	Hex	Dec	Binary	Char
w	07	7	00000111	.
tmr0	00	0	00000000	.
option_reg	07	7	00000111	.
pcl	2B	43	00101011	+
pclath	00	0	00000000	.
status	3C	60	00111100	<
fsr	00	0	00000000	.
porta	00	0	00000000	.
trisa	3F	63	00111111	?
portb	00	0	00000000	.

I registri che si modificano nell'esecuzione vengono evidenziati con il colore rosso.

Compilazione

Se selezioniamo l'opzione di compilazione, Project → Build All, o clicchiamo il pulsante della barra degli strumenti, vedremo che il programma che abbiamo presentato come soluzione, si assembla senza errore. Appariranno cinque messaggi per avvertirci di verificare che i registri indicati siano nei banchi di memoria definiti nel programma. Questi messaggi non indicano degli errori, semplicemente avvisano di una situazione che al compilatore risulta strana.

Nel caso abbiate realizzato un programma diverso da quello presentato e il processo di compilazione produca degli errori, sarà necessario correggerli linea per linea, fino ad ottenere un programma libero da errori.

Simulazione

Il fatto che un programma non dia errori di compilazione è un passo importante verso la soluzione finale, però non è quello definitivo, infatti ci potrebbero essere degli errori per quanto riguarda il funzionamento che ci aspettiamo. Per evitare questo MPLAB ci offre la possibilità di simulare il programma ed è esattamente quello che faremo.

Simulazione di un ingresso

Quando abbiamo studiato MPLAB abbiamo visto tutte le possibilità della simulazione, eccetto quella di simulare uno o più ingressi. In questo programma dipendiamo dal valore di un ingresso (RA0) per poter verificare la risposta del sistema, però questo ingresso si può simulare ed è ciò che impareremo a fare. Selezionando Debug → Simulator Stimulus si apre un menù come quello della figura dove sono presentate quattro opzioni:

– Asynchronous Stimulus (Stimolo asincrono): apre un quadro di dialogo interattivo per controllare i segnali di ingresso sui pin di ingresso del microcontroller.

– Stimulus Pin File (File di stimoli sui terminali): mediante un file di testo si descrivono i segnali di ingresso sui pin.

– Stimulus Register File (File di registro degli stimoli): si utilizza il contenuto di un file di testo per porre direttamente a 1 oppure a 0 ognuno degli 8 bit di un registro.

– Clock Stimulus (Segnale di clock): genera un'onda quadra programmabile.

```
MPLAB IDE - C:\PROGRA~1\MPLAB\PROGETTI\ED28.PJT
File Project Edit Debug PICSTART Plus Options Tools Window Help

;segmenti collegato alla porta B si visualizzeranno sequenzialmente i numeri da 1 a
;e su i LED collegati alla porta C lo stesso numero però in binario. Questo avviene
;a intervalli di 0,05". Quando RA0 passa a livello "0", visualizziamo il numero casu
;ottenuto per un tempo di 3". Dopo, il display e i LED si spengono e la sequenza si

                List    p=16F870           ;Processore
                include "P16F870.INC"      ;Definizioni dei registri interni

Numero          equ     0x20               ;Numero casuale
Delay_Cont      equ     0x21               ;Contatore di intervalli
Temporale       equ     0x22               ;Variabile temporale

                org     0x00               ;Vector di Reset
                goto    Inizio
                org     0x05               ;Salva il vector di interrupt
;-----------------------------------------------------------------------------------
;Tabella: Routine che converte il codice binario presente sui 4 bit meno significati
;del registro W nel loro equivalente a 7 segmenti. Il codice a 7 segmenti ritorna su

Tabella:        addwf   PCL,F              ;Spostamento sulla tabella
                retlw   b'00111111'        ;Numero 0
                retlw   b'00000110'        ;Numero 1
                retlw   b'01011011'        ;Numero 2
                retlw   b'01001111'        ;Numero 3
                retlw   b'01100110'        ;Numero 4
                retlw   b'01101101'        ;Numero 5
```

SFR Name	Hex	Dec	Binary	Char
w	07	7	00000111	.
tmr0	15	21	00010101	.
option_reg	FF	255	11111111	.
pcl	00	0	00000000	.
pclath	00	0	00000000	.
status	1C	28	00011100	.
fsr	00	0	00000000	.
porta	00	0	00000000	.
trisa	3F	63	00111111	?
portb	00	0	00000000	.
trisb	FF	255	11111111	.
portc	00	0	00000000	.
trisc	FF	255	11111111	.
intcon	00	0	00000000	.
pir1	00	0	00000000	.
pie1	00	0	00000000	.
pir2	00	0	00000000	.
pie2	00	0	00000000	.
tmr1l	00	0	00000000	.
tmr1h	00	0	00000000	.
t1con	00	0	00000000	.
tmr2	00	0	00000000	.
pr2	FF	255	11111111	.
t2con	00	0	00000000	.
ccpr1l	00	0	00000000	.
ccpr1h	00	0	00000000	.
ccp1con	00	0	00000000	.
rcsta	00	0	00000000	.

Watch_1

Address	Symbol	Value
20	Numero	H'05'
200	w	H'07'
22	Temporale	H'09'
21	Delay_Cont	H'00'
06	portb	H'00'
07	portc	H'00'

Modify — Address: / End Address: / Data/Opcode: / Fill Range / Read Range / Radix: Hex, Decimal / Auto Increment / Memory Area: Data, Program, Stack, EEPROM / Write / Read / Close / Help

Asynchronous Stimulus Dialog — RA0 (T) | Stim 2 (P) | Stim 3 (P) | Stim 4 (P) | Stim 5 (P) | Stim 6 (P) | Stim 7 (P) | Stim 8 (P) | Stim 9 (P) | Stim 10 (P) | Stim 11 (P) | Stim12 (P) / Close / Help

Ln 17 Col 1 | 139 | WR | No Wrap | INS | PIC16F870 | pc:0x00 | w:0x07 | --Z dc c | Bk On | Sim | 4 MHz | User

Ecco come appare la videata dopo aver creato le finestre di simulazione.

Quando selezioniamo la prima opzione, stimolo asincrono, appare la finestra della figura. La finestra ha dodici pulsanti e a ognuno di essi possiamo assegnare una linea di ingresso, quindi potremo simulare dodici ingressi. Per assegnare un ingresso a un pulsante dobbiamo cliccare il pulsante destro del mouse sul pulsante desiderato.

Così facendo si apre un menù come quello della figura della pagina precedente. Mediante questo tipo di menù possiamo scegliere il tipo di stimolo, ovvero la forma che avrà il segnale di ingresso ogni volta che attiveremo il pulsante: un solo impulso (Pulse), livello basso o 0 (Low), livello alto o 1 (High), o scambiare fra di loro i livelli quando si clicca il pulsante (Toggle). Per assegnare l'ingresso al pulsante selezioneremo "Assign Pin" aprendo così un nuovo menù in cui troviamo la lista di tutti i pin di ingresso del microcontroller.

Selezioneremo RA0 e il modo di stimolo come alternato o Toggle, perciò quando cliccheremo il pulsante per la prima volta l'ingresso passerà a livello alto sino a quando non attiveremo di nuovo il pulsante.

Lista dei pin a cui è possibile assegnare il pulsante.

Watch_1

Address	Symbol	Value
20	Numero	H'07'
200	w	H'3C'
22	Temporale	H'09'
21	Delay_Cont	H'01'
06	portb	H'67'
07	portc	H'09'

Risultato della prima sequenza con i valori sulle porte di uscita.

Modify

Address: PCL
End Address:
Data/Opcode: 6
Fill Range
Read Range
Radix: Hex Decimal
Auto Increment
Memory Area: Data Program Stack EEPROM
Write Read Close Help

Forziamo un valore sul registro contatore di programma.

Finestre di simulazione

Per eseguire la simulazione del programma conviene aprire la finestra dei Registri delle Funzioni Speciali (Special Function Registers). Vi consigliamo inoltre di creare una finestra per poter visualizzare lo stato dei registri più importanti: variabili, registro di lavoro e porte di uscita. Nello stesso tempo avremo una finestra di creazione di stimoli per forzare gli ingressi, e infine dovremo disporre della finestra Modify per poter forzare il valore di un registro ed evitare che il programma rimanga bloccato in cicli infiniti; come capiterebbe ad esempio nella routine di temporizzazione. Nella figura della pagina precedente possiamo vedere che aspetto avrà il video.

Simulazione del programma

Per iniziare la simulazione passo a passo dovremo cliccare l'icona o premere il tasto F7. In questo modo vedremo come il programma inizia la sua esecuzione configurando come prima cosa i dispositivi del PIC. Potremo vedere nella finestra dei registri delle funzioni speciali, come si modificano i valori dei registri lungo il corso dell'esecuzione. Quando si modifica un registro, questo verrà evidenziato in rosso.

Il programma si fermerà in un ciclo (Loop) aspettando di ricevere l'ordine di inizio, cioè che si attivi l'ingresso, quindi per simulare questo dovremo ricorrere al nostro generatore di stimoli e cliccare il pulsante che abbiamo assegnato a RA0. Così facendo, sarà come dire al simulatore che è arrivato un valore 1 (questo valore non cambierà fino a quando non cliccheremo nuovamente il pulsante di stimolo) salteremo l'istruzione "goto Loop" continuando con il programma. Abbiamo acquisito un valore casuale e passiamo alla routine di temporizzazione.

Qui capiterà la stessa cosa di prima, rimarremo in attesa fino a quando il flag di overflow del Timer 0 non passerà a 1. Per evitare un ciclo infinito useremo la finestra Modify selezionando il flag T0IF e scrivendo come valore un 1. Facendo questo salteremo alla tabella di conversione, però non potremo muoverci su questa tabella se non avremo un valore corretto sul contatore di programma, quindi gli assegniamo un valore, ad esempio 6, come mostrato nell'immagine.

Il programma esce dalla tabella e adatta il numero acquisito, il quale, mediante sottrazioni consecutive avrà un valore fra 1 e 9. Ripeteremo il ciclo fino a ottenere tale risultato e a partire da questo punto verranno presentati i numeri in modo sequenziale o il numero casuale in funzione dello stato di RA0.

Conclusioni

Questo esercizio completo non ha nessuna funzionalità se non si impara a giocare con esso. Modificando il programma, provate tutto ciò che è possibile, verificate i vostri dubbi e imparate a gestire il simulatore; riassumendo, è molto importante non avere alcun timore e dedicare il tempo necessario a imparare e prendere confidenza con quanto appreso.